- 8 继承
  - 概述
    - 1 . 多个类中存在相同的属性和行为是,将这些内容抽取到单个类中,那么多个类无需再定义属性和行为,只需继承即可
    - 2 . 多个类称为子类或派生类,单个类称为父类或者超类
    - 3 . 通过extends关键字让他们产生关系
  - 好处
    - 1 . 提高了代码的复用性
    - 2 . 让类与类产生关系,是多态的前提
  - 特点
    - 1 . java中的类只支持单继承,不支持多继承(因为如果可以多继承,就会出现调用不明确的问题)
    - 2 . java支持多层继承(继承体系)
  - 何时使用
    - 子类是父类的一种,若果B类是A类的一种就使用继承关系
    - 注意 : 不要为了获取某个功能而去继承
  - 子父类中变量查找
    - 1 . 首先在局部范围找,有就使用
    - 2 . 在本类成员位置找,有就使用
    - 3 . 在父类成员位置找, 有就使用,没有就报错
  - this与super
    - 概念上
      - super代表的是父类内存空间的标识
      - this代表所在方法所属对象的引用
    - 使用时机
      - 1 . 当子父类中出现同名成员可以用super区分
      - 2 . 子类调用父类的构造方法可以用super
    - 调用方式
      - super
        - 通过super可以直接调用父类的构造方法,成员变量,成员方法
        - 调用构造方法 : super()
        - 调用成员变量 : super .变量名
        - 调用成员方法 : super.方法名();
      - this
        - 通过this可以直接调用本类的构造方法,成员变量,成员方法
        - 调用构造方法 : this()
        - 掉用成员变量 : this.变量名
        - 调用成员方法 : this.方法名();
    - 注意
      - 1 . this代表本类对象的引用
      - 2 . super代表父类内存空间的标识
  - 重写override和重载overload
    - 重写
      - 概念
        - 子父类关系中,子类中存在与父类中方法声明完全一样的方法叫重写(即 : 除了方法体以外其他都相同,但是子类修饰符权限要大于等于父类)
      - 作用
        - 当子类需要父类功能,而功能主题有自己特有的内容时,可以复写父类中的方法,这样既沿袭了父类功能,又定义类子类特有内容
        - 举例 : 父亲吧武功传给儿子, 儿子可以进行改进,或者不改
      - 注意 :
        - 只有方法才有重写,变量不存在
    - 重载
      - 概念
        - 一个类中或者子父类关系中存在多个同名方法,并且参数列表不同的方法叫重载(子父类关系中 : 子类中的方法和父类中的方法构成重载关系)
  - 子父类间成员方法注意事项
    - 1 . 父类中私有的方法不能被覆盖(因为私有的不能被继承,所以不算重写)
    - 2 . 子类方法访问权限大于父类方法访问权限(父类如果会死public,子类只能是public)
  - 子父类构造方法
    - 特点
      - 子类默认构造方法访问了父类的无参构造方法,子类构造方法中都有默认语句super(); , 并且默认只能是无参的super();语句
      - 因为子类会具备父类中非私有数据,这个初始化靠父类完成,所以先调用父类构造方法来初始化父类中的数据
    - 如果父类无无参构造,子类构造如何访问父类构造
      - 1 . 通过super(参数)调用父类带参构造
      - 2 . 通过this(..)调用本类的构造方法 (如果一个类中所有构造第一句都调用this(..)这样是错误)
    - 构造方法执行完后会创建该类对象吗?
      - 不一定,子类的构造方法第一行调用父类的构造方法是,不会创建对象
    - 注意
      - 构造方法中的this()和super()语句必须放在第一行,并且只能取其一
  - 子父类间代码块和构造方法执行顺序
    - 1 . 优先级 : 静态代码块---->构造代码块---->构造方法
    - 2 . 执行顺序 : 父类静态代码块---->子类静态代码块---->父类构造代码块---->父类构造方法---->子类构造代码块---->子类构造方法
    - 静态代码块作用 : 初始化静态成员变量
    - 构造代码块作用:
      - 1 . 初始化非静态成员变量
      - 2 . 把所有构造方法中的共性内容抽取出来,提高代码的复用性